

UNITED STATES PATENT APPLICATION
FOR
TRANSMITTING AND DISPLAYING COMPOSITE IMAGES

INVENTORS:

Jordan M. Slott, a citizen of the United States
Thomas G. O'Neill, a citizen of the United States

ASSIGNED TO:

Sun Microsystems, Inc., a Delaware Corporation

PREPARED BY:

THELEN, REID & PRIEST LLP
P.O. BOX 640640
SAN JOSE, CA 95164-0640
TELEPHONE: (408) 292-5800
FAX: (408) 287-8040

Attorney Docket Number: SUN-030102

Client Docket Number: 811173-409

SPECIFICATIONTITLE OF INVENTION

TRANSMITTING AND DISPLAYING COMPOSITE IMAGES

CROSS-REFERENCE TO RELATED APPLICATION

[0001] The present application is a continuation-in-part of Application Serial No. 10/449,540, filed on May 29, 2003, by Jordan M. Slott and Thomas G. O'Neill, entitled "Caching Backing-Store Images To Reduce Bandwidth", (Attorney Docket No. SUN-P9020), and of Application Serial No. 10/385,759, filed on March 10, 2003, by Thomas G. O'Neill and Jordan M. Slott, entitled, "Tagging Repeating Images For Improved Compression", (Attorney Docket No. SUN-P9199), which is a continuation-in-part of co-pending Application Serial No. 10/247,907, filed on September 20, 2002, by Thomas G. O'Neill and Jordan M. Slott, entitled "Tagging Single-Color Images For Improved Compression", (Attorney Docket No. SUN-P7083), which is a continuation in part of co-pending Application Serial No. 10/194,123, filed on July 11, 2002, by Thomas G. O'Neill and Jordan M. Slott, entitled "Tagging Multicolor Images For Improved Compression", (Attorney Docket No. SUN-P7082), and of Application Serial No. _____, filed on July 17, 2003, by Jordan M. Slott and Thomas G. O'Neill, entitled "Caching Fonts For Improved Bandwidth of Transmitted Text", (Attorney Docket No. SUN-P7932).

FIELD OF THE INVENTION

[0002] The present invention relates to the field of computer graphics. More particularly, the present invention relates to the caching of fonts in order to improve bandwidth of transmitted text.

BACKGROUND OF THE INVENTION

[0003] Remote computing applications where screen information is generated on one computer (a "host") and transmitted for display on another computer ("a display") are growing in popularity. Examples of some display computers include multipurpose PCs, thin-clients, and Personal Digital Assistants (PDAs).

[0004] The X Windows System is a standard that utilizes X client software and X server software to enable the updating of displays at the requests of X client applications. The X server software runs on the host computer. An X client application, which may be running on a different computer, communicates with the X server software by utilizing a low-level library of routines known as Xlib. Xlib provides the functions required to connect to display servers, create windows, render graphics, respond to events, etc. The X server software then may interface with display drivers to actually render the graphics on the display.

[0005] X is frequently used as a "local display application", where the X-server and display are both on the same computer. That is, the same computer is both the "host" and "display" computer. Examples of this usage include running an X-server on a workstation or on an X-

terminal. An X-terminal typically has a computer processor, graphics subsystem and display, but no hard drive. Applications running on other computers use the Xlib routines to communicate with the X-server software running on the X-terminal.

[0006] While in some contexts it is advantageous to have the X server and the display on the same computer, this is not necessarily the case in other contexts. One specific context that will be discussed is a remote display application. In such a design, client applications make requests of a centralized server or servers (here known collectively as the "host computer"). The host computer then manages one or more "display computers", which are typically simple terminal devices.

[0007] The Sun Ray™ appliance from Sun Microsystems, Inc. of Palo Alto, CA is an example of a thin client which serves as a "display computer" in a remote computing application. A Sun Ray™ appliance has a processor, graphics subsystem and display, but no hard drive. A Sun Ray™ appliance is a "display" computer and runs its own proprietary software. The Sun Ray™ server is the "host" computer and runs the X-server software. The full screen image is maintained both in RAM on the host computer as well as in the frame buffer of the Sun Ray™ appliance's video card. In order to synchronize the displays, the host computer sends screen update information to the Sun Ray™ appliance via a network protocol known as NewT. The Sun Ray™ appliance uses the protocol commands to update the state of its hardware frame buffer.

[0008] To achieve the appearance of an instantaneous update, graphic application programs frequently construct a composite image off-screen and then blit the composite image onto the screen. For example, in X Windows, an X client application may build the composite image in an off-screen pixmap and then copy it to a realized window using XCopyArea. In many thin client architectures, a large composite image will require more than one protocol command to send. At low bandwidth, the composite image will appear gradually instead of instantaneously - defeating the intention of the application programs and degrading the user experience.

[0009] What is needed is a solution for reducing the perceived lag experienced by a user in such systems.

BRIEF DESCRIPTION

[0010] A solution is provided that transmits and displays a composite image in a way that reduces the lag perceived during the drawing of the composite image on a screen. The host computer tell the display computer when the composite image is sent. The display computer may store the image in an off-screen memory, and then write the entire contents of the off-screen memory on to the on-screen display at once, in a process commonly known as blitting. This can dramatically reduce the perceived lag caused by the time it takes to transmit the composite image over the network. In an alternative embodiment, the display computer may buffer the protocol commands for the composite image rather than executing them as they arrive. When all of the commands for the image have arrived, the display computer may execute them as rapidly as possible to give the appearance of a near-instantaneous update.

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] The accompanying drawings, which are incorporated into and constitute a part of this specification, illustrate one or more embodiments of the present invention and, together with the detailed description, serve to explain the principles and implementations of the invention.

[0012] In the drawings:

FIG. 1 is a block diagram illustrating a remote display application such as a Sun Ray™ network.

FIG. 2 is a block diagram illustrating a remote display application in accordance with an embodiment of the present invention.

FIG. 3 is a diagram illustrating blitting in accordance with an embodiment of the present invention.

FIG. 4 is a flow diagram illustrating a method for transmitting a composite image to a display computer in accordance with an embodiment of the present invention.

FIG. 5 is a diagram illustrating the maximum available memory example in accordance with an embodiment of the present invention.

FIG. 6 is a diagram illustrating the equal size example in accordance with another embodiment of the present invention.

FIG. 7 is a diagram sending the protocol commands used to draw a composite image in accordance with an embodiment of the present invention.

FIG. 8 is a flow diagram illustrating a method for sending a composite image from a host computer to a display computer in accordance with an embodiment of the present invention.

FIG. 9 is a flow diagram illustrating a method for handling a composite image sent from a host computer to a display computer in accordance with an embodiment of the present invention.

FIG. 10 is a flow diagram illustrating a method for handling a composite image sent from a host computer to a display computer in accordance with another embodiment of the present invention.

DETAILED DESCRIPTION

[0013] Embodiments of the present invention are described herein in the context of a system of computers, servers, and software. Those of ordinary skill in the art will realize that the following detailed description of the present invention is illustrative only and is not intended to be in any way limiting. Other embodiments of the present invention will readily suggest themselves to such skilled persons having the benefit of this disclosure. Reference will now be made in detail to implementations of the present invention as illustrated in the accompanying drawings. The same reference indicators will be used throughout the drawings and the following detailed description to refer to the same or like parts.

[0014] In the interest of clarity, not all of the routine features of the implementations described herein are shown and described. It will, of course, be appreciated that in the development of any such actual implementation, numerous implementation-specific decisions must be made in order to achieve the developer's specific goals, such as compliance with application- and business-related constraints, and that these specific goals will vary from one implementation to another and from one developer to another. Moreover, it will be appreciated that such a development effort might be complex and time-consuming, but would nevertheless be a routine undertaking of engineering for those of ordinary skill in the art having the benefit of this disclosure.

[0015] In accordance with the present invention, the components, process steps, and/or data structures may be implemented using various types of operating systems, computing platforms, computer programs, and/or general purpose machines. In addition, those of ordinary skill in the

art will recognize that devices of a less general purpose nature, such as hardwired devices, field programmable gate arrays (FPGAs), application specific integrated circuits (ASICs), or the like, may also be used without departing from the scope and spirit of the inventive concepts disclosed herein.

[0016] In the present application, an embodiment of the present invention is described in the context of a modification of traditional X Windows technology for use with remote computing applications. However, one of ordinary skill in the art will recognize that other embodiments are possible and the present invention should not be limited to X Windows, Sun Ray™, or remote computing technology.

[0017] In remote display applications based on an X server, the Device Dependent X (DDX) layer of the X server may be modified to include possible compression of the screen data and transmission of remote display protocol commands to the display computer. The DDX provides a software interface to a conceptual hardware device. In a specific embodiment of the present invention, the desired results can be accomplished by modifying the DDX, without changing the DIX.

[0018] FIG. 1 is a block diagram illustrating a remote display application such as a Sun Ray™ network. Client applications 100a-100c send information to and receive information from the host computer 102 over the X Windows Protocol. Some of the information comprises composite images, wherein multiple drawing commands are used to draw a single screen image. The host computer utilizes a device independent layer (DIX) 104 to act as a controlling layer,

which handles initialization procedures and all device-independent handling of requests and events. A device dependent layer (DDX) 106 then is used to handle device dependent functionality. The DDX 106 then communicates with display computer 108 using the NewT protocol. The DIX and the DDX essentially make up an X-server.

[0019] FIG. 2 is a block diagram illustrating a remote display application in accordance with an embodiment of the present invention. Client applications 200a-200c send information to and receive information from the host computer 202 over the X Windows Protocol. Some of the information is composite image information. The host computer utilizes a device independent layer (DIX) 204 to act as a controlling layer, which handles initialization procedures and all device-independent handling of requests and events. A device dependent layer (DDX) 206 then is used to handle device dependent functionality. An buffering management layer 208 may then be provided which performs the additional buffering management techniques discussed in the present application. Alternatively, this buffering management layer may simply be part of a modified DDX. The buffering management layer 208 then communicates with display computer 210 using an enhanced version of the NewT protocol.

[0020] The present invention provides a mechanism that transmits a composite image in a way that reduces the perceptible lag between drawing the portions of the image on a screen. Each display computer may be equipped with an off-screen memory. The host computer may tell the display computer when a composite image is being sent. The display computer may use the protocol commands for the composite image to reconstruct the composite image in the off-screen memory instead of directly on screen. When the composite image is complete, the

display computer may transfer the composite image from the off-screen memory on to the screen in a process known as blitting. As will be seen, this results in the ability to dramatically reduce the perceived lag due to the time it takes to transmit the composite image over the network

[0021] FIG. 3 is a diagram illustrating blitting in accordance with an embodiment of the present invention. Data is transmitted from the host computer to the off-screen memory 300. The pixel data in off-screen memory may then be copied all at once to the display 302.

[0022] FIG. 4 is a flow diagram illustrating a method for transmitting a composite image to a display computer in accordance with an embodiment of the present invention. At 400, the host computer may determine the maximum amount of memory available on the display computer for building the composite images in the off-screen memory. This memory may come from, for example, main memory or dedicated video memory. In one embodiment, the display computer may communicate the maximum amount of memory available for such purposes (i.e., indicates a size of memory available) and optionally any stride requirements. This information may be communicated either at boot time (when the connection between the display computer and host computer is established), or at some time subsequent to boot time.

[0023] When the host computer needs to transmit a composite image to the display computer, best results will be obtained if the display computer has enough off-screen memory to store the entire image. In this case the screen update will appear to occur instantaneously on the display computer. If the display computer does not have enough available memory for the entire composite image, the host computer may break up the single composite image into multiple

composite images at 402. The host computer must then decide how it wishes to do this, however each sub-image should fit into the off-screen memory. In one embodiment it may break up the image into one or more rectangles to fit the maximum available memory and then transmit one or more smaller remainder regions for any area left-over at the end. FIG. 5 is a diagram illustrating the maximum available memory example in accordance with an embodiment of the present invention. If, for instance, off-screen memory 502 can contain 30,000 pixels, then a composite image 500 of size 400 x 350 may be broken into four rectangles 504, 506, 508, 510 of size 400 x 75, as well as a remainder rectangle 512 of size 400 x 50.

[0024] In another embodiment, it may break up the image into rectangular areas of a fixed, equal size. FIG. 6 is a diagram illustrating the fixed, equal size example in accordance with another embodiment of the present invention. Here the same 400 x 350 composite image 600 of FIG. 5 may instead be split into fixed-size rectangles of 100 x 100, each of which fit in off-screen memory 602. The use of rectangles of size 100 x 100 is just an example and different size rectangles may be chosen as long as they fit in the off-screen memory. The 400 x 350 composite image 500 may be split into rectangles 604, 606, 608, 610, 612, 614, 616, 618, 620, 622, 624, 626 of size 100 x 100 as well as remainder rectangles 628, 630, 632, 634 of size 100 x 50.

[0025] Returning to FIG. 4, when transmitting a composite image from the host to the display computer, it will transmit one sub-image of the composite image at a time. Therefore, at 404 it may determine if there are any more rectangles to send. If not, the process ends. If so, then at 406 it may first send a message telling the display computer that a composite image is about to be sent. At this point, it may also tell the display computer the dimensions of the

composite image. Then at 408, it may send the sub-image to the display computer. The display computer may store the sub-image in the off-screen memory. When the host computer has finished transmitting the sub-image, it may tell the display computer that it has finished and give the destination on-screen coordinates of the sub-image. The destination on-screen coordinates of the sub-image may be given in any of the messages transmitted from the host computer to the display computer, however. The display computer may then copy the sub-image from its off-screen memory on to the screen.

[0026] In an embodiment of the present invention, in the context of the X-server, the process may be executed when, for example, a CopyArea command is used to copy a pixmap to an on-screen window. When the CopyArea command is executed, the host computer may tell the display computer a composite image is about to sent along with the dimensions of the pixmap. When finished, the host computer may tell the display computer the destination on-screen coordinates. Other X commands which copy composite images to realized windows, such as RestoreAreas, may be handled in the same manner.

[0027] While CopyArea is a specific command in the X Windows System, the term copy area command may be used in the present document to refer to any rendering command used to copy a pixmap on to the screen.

[0028] While RestoreAreas is a specific command in the X Windows System, the term restore areas command may be used in the present document to refer to any rendering command

used to restore an obscured area of the screen stored in a backing-store mechanism on to the screen.

[0029] A pixmap is an array of pixel values in memory which can be used to store an image off-screen on the host computer. Each pixel is represented by a certain number of bits depending upon the number of possible colors in the image. On the one hand, a single bit-per-pixel bitmap is used to represent up to two color images. On the other hand, a 24 bits-per-pixel bitmap is used for images with many colors.

[0030] While pixmap is a specific term in the X Windows system, the term pixmap may be used in the present document to refer to any off-screen image.

[0031] The present invention may be used in conjunction with "region tagging" concepts. In these concepts, composite images are typically sent via a number of protocol commands, each describing a relatively small portion of the composite image (e.g., a text protocol command may describe a single line of text in an entire web page displayed by a web browser). Use of the present invention reduces the appearance of distracting piece-meal updates at lower bandwidths. Additionally, when using an "underlay embodiment", a "flashing artifact" may occur if the same composite image is copied to the screen more than once. The multiple copies may occur, for example, due to a bug in the application or an inefficiency in the underlying graphical toolkit. The artifact is caused by the background color being drawn over other parts of the composite, and the present invention reduces or eliminates this problem.

[0032] FIG. 7 is a diagram illustrating another embodiment of the present invention. In this embodiment, rather than sending sub-images of the composite image, the protocol commands used to draw the entire composite image are sent and buffered on the display computer. The host computer may determine the maximum amount of memory available on the display computer for buffering the protocol commands. In one embodiment, the display computer may communicate the maximum amount of memory available for such purposes (i.e., indicate a size of memory available). A start command 700 may indicate the beginning of a composite image. This command, as well as all the protocol commands used to draw the composite image 702 may be transmitted in a stream to the display computer, followed by an end command 704. The display computer may store the commands in a buffer 706 and, once the end command is received, execute the drawing commands in order from the buffer, drawing the appropriate regions directly on-screen. If the commands are executed quickly enough, this procedure will give the appearance of an instantaneous update. Because the protocol commands will typically occupy less memory than the entire composite image, this method may yield better results when the display computer does not have enough room for the entire composite image off-screen.

[0033] FIG. 8 is a flow diagram illustrating a method for sending a composite image from a host computer to a display computer in accordance with an embodiment of the present invention. The display computer may have an off-screen memory with available memory. At 800, the amount of memory available in the off-screen memory may be determined. This may be determined by receiving an indicator of the amount of memory available in the off-screen memory from the display computer when the connection between the display computer and the host computer is established, or any time thereafter. Alternatively, it may be determined by

receiving an indicator of a width and height and stride requirements of memory available in the off-screen memory from the display computer when the connection between the display computer and the host computer is established, or any time thereafter.

[0034] At 802, the composite image may be broken into one or more sub-images, wherein each of the sub-images can fit into the available memory of the off-screen memory. Each of the sub-images may be a rectangle. This may include breaking the composite image into sub-images that are chosen so as to maximize the number of sub-images that are equal in size to the available memory in the off-screen memory. Alternatively, this may include breaking the composite image into sub-images that are chosen so as to be equal in size to each other. Each of the sub-images may be transmitted to the display computer for storage in the off-screen memory. This may include sending one or more messages indicating that a sub-image of a composite image is being transmitted and containing data sufficient to reconstitute the sub-image on-screen. For each of the sub-images, this may include the following. At 804, a message may be sent to the display computer indicating that a sub-image of a composite image is about to be transmitted. At 806, a message may be sent to the display computer indicating the width and height of the sub-image may be sent. At 808, a message may be sent indicating the position on the screen to which the sub-image of the composite image is to be displayed. At 810, the sub-image may be transmitted. This may require more than one protocol command. At 812, a message may be sent to the display computer indicating the transmission of the sub-image has been completed.

[0035] FIG. 9 is a flow diagram illustrating a method for handling a composite image sent from a host computer to a display computer in accordance with an embodiment of the present

invention. The display computer may have an on-screen display with an off-screen memory having available memory. At 900, an indicator of the amount of memory available in the off-screen memory may be sent to the host computer when the connection between the display computer and the host computer is established, or any time thereafter. Alternatively, an indicator of a width and height and stride requirements of memory available in the off-screen memory may be sent to the host computer when the connection between the display computer and the host computer is established, or any time thereafter.

[0036] One or more sub-images of the composite image may then be received from the host computer. Each of the sub-images may be a rectangle. The receiving may include receiving one or more messages from the host computer indicating a sub-image of a composite image is being transmitted, wherein the one or more messages comprise data sufficient to reconstitute the sub-image on screen. For each of the sub-images, this may include the following. At 902, a message may be received from the host computer indicating that a sub-image of a composite image is about to be transmitted. At 904, a message may be received from the host computer indicating the width and height of the sub image. At 906, a message may be received indicating the position on the screen to which the sub-image of the composite image is to be displayed. At 908, the sub-image may be received. At 910, the sub-image may be stored in the off-screen memory. At 912, a message may be received from the host computer indicating the transmission of the sub-image has been completed. At 914, the display computer may transfer the sub-image from the off-screen memory to the on-screen display.

[0037] FIG. 10 is a flow diagram illustrating a method for handling a composite image sent from a host computer to a display computer in accordance with another embodiment of the present invention. The display computer may have a buffer and an on-screen display. At 1000, a message may be received from the host computer indicating that the composite image is about to be transmitted. At 1002, one or more protocol commands used to draw the composite image may be received. At 1004, the one or more protocol commands may be stored in the buffer. At 1006, a message from the host computer indicating that the transmission of the composite image has been completed may be received. At 1008, the composite image may be drawn on the on-screen display by executing the one or more protocol commands in the buffer. For best results, the processing of the protocol commands should occur as quickly as possible.

[0038] FIG. 11 is a block diagram illustrating an apparatus for sending a composite image from a host computer to a display computer in accordance with an embodiment of the present invention. The display computer may have an off-screen memory with available memory. An amount of available off-screen memory determiner 1100 may determine the amount of memory available in the off-screen memory. This may be determined by receiving an indicator of the amount of memory available in the off-screen memory from the display computer when the connection between the display computer and the host computer is established, or any time thereafter. Alternatively, it may be determined by receiving an indicator of a width and height and stride requirements of memory available in the off-screen memory from the display computer when the connection between the display computer and the host computer is established, or any time thereafter.

[0039] A composite image sub-image divider 1102 coupled to the amount of available off-screen memory determiner 1100 may break the composite image into one or more sub-images, wherein each of the sub-images can fit into the available memory of the off-screen memory. Each of the sub-images may be a rectangle. This may include breaking the composite image into sub-images that are chosen so as to maximize the number of sub-images that are equal in size to the available memory in the off-screen memory using a maximum number of sub-images equal in size to available memory sub-image divider 1104. Alternatively, this may include breaking the composite image into sub-images that are chosen so as to be equal in size to each other using a fixed, equal size sub-image divider 1106. Each of the sub-images may be transmitted to the display computer for storage in the off-screen memory using a sub-image transmitter 1108 coupled to the composite image sub-image divider 1102. This may include sending one or more messages indicating that a sub-image of a composite image is being transmitted and containing data sufficient to reconstitute the sub-image on-screen using a sub-image one or more message sender 1110. For each of the sub-images, this may include the following. An impending sub-image transmission message sender 1112 may send a message to the display computer indicating that a sub-image of a composite image is about to be transmitted. A sub-image size message sender 1114 may send a message to the display computer indicating the size of the sub image. A sub-image stride requirements message sender 1116 may optionally send a message to the display computer indicating any stride requirements of the sub-image. A sub-image screen position message sender 1118 may send a message indicating the position on the screen to which the sub-image of the composite image is to be displayed. A sub-image sender 1120 coupled to the sub-image screen position message sender 1118 may transmit the sub-image. Finally, a sub-image transmission completed message sender 1122 coupled to the sub-image sender 1120 may

send a message to the display computer indicating the transmission of the sub-image has been completed.

[0040] FIG. 12 is a block diagram illustrating an apparatus for handling a composite image sent from a host computer to a display computer in accordance with an embodiment of the present invention. The display computer may have an on-screen display with an off-screen memory having available memory. An available off-screen memory size sender 1200 may send an indicator of the amount of memory available in the off-screen memory to the host computer when the connection between the display computer and the host computer is established, or any time thereafter. Alternatively, an available off-screen memory width and height and stride requirement sender 1202 may send an indicator of a width and height and stride requirements of memory available in the off-screen memory to the host computer when the connection between the display computer and the host computer is established, or any time thereafter.

[0041] One or more sub-images of the composite image may then be received from the host computer using a sub-image receiver 1204 coupled to the available off-screen memory size sender 1200. Each of the sub-images may be a rectangle. The receiving may include receiving one or more messages from the host computer indicating a sub-image of a composite image is being transmitted, wherein the one or more messages comprise data sufficient to reconstitute the sub-image on screen, using a sub-image one or more message receiver 1206. For each of the sub-images, this may include the following. An impending sub-image transmission message receiver 1208 may receive a message from the host computer indicating that a sub-image of a composite image is about to be transmitted. A sub-image width and height message receiver

1210 may receive a message from the host computer indicating the width and height of the sub image. A sub-image screen position message receiver 1212 may receive a message indicating the position on the screen to which the sub-image of the composite image is to be displayed. A sub-image off-screen memory storer 1214 coupled to the sub-image receiver 1204 may receive one or more messages specifying the sub-image image data and store the sub-image in the off-screen memory. A sub-image transmission completed message receiver 1216 coupled to the sub-image screen position message receiver 1212 may receive a message from the host computer indicating the transmission of the sub-image has been completed. A sub-image off-screen memory-to-one-screen display transferrer 1218 coupled to the sub-image off-screen memory storer 1214 may transfer the sub-image from the off-screen memory to the on-screen display.

[0042] FIG. 13 is a block diagram illustrating an apparatus for handling a composite image sent from a host computer to a display computer in accordance with another embodiment of the present invention. The display computer may have a buffer 1300 and an on-screen display 1302. An impending composite image transmission message receiver 1304 may receive a message from the host computer indicating that the composite image is about to be transmitted. A protocol command receiver 1306 coupled to the impending composite image transmission message receiver 1304 may receive one or more protocol commands used to draw the composite image. A protocol command buffer storer 1308 coupled to the protocol command receiver 1306 and to the buffer 1300 may store the one or more protocol commands in the buffer. A composite image transmission completed message receiver 1310 coupled to the protocol command receiver 1306 may receive a message from the host computer indicating that the transmission of the composite image has been completed. A composite image on-screen display executor 1312

coupled to the buffer 1300 and to the display 1302 may draw the composite image on the on-screen display by executing the one or more protocol commands in the buffer.

[0043] While embodiments and applications of this invention have been shown and described, it would be apparent to those skilled in the art having the benefit of this disclosure that many more modifications than mentioned above are possible without departing from the inventive concepts herein. The invention, therefore, is not to be restricted except in the spirit of the appended claims.